



Sie kommt!: Die erste Ausgabe im Jahr 2013

Herausgeber und für den Inhalt verantwortlicher:
Leszek Chmielewski, Prager Straße 92/11/12, 1210 Wien, Österreich
@Mail: retrozx@gmail.com
Klubkonto (Inhaber: Bernhard Lutz):IBAN: DE59 5486 2500 0000 5461 43
SWIFT-Code: GENODE6K, BIC-Code: GENODE61SUW
KTO.: 546143, BLZ: 54862500 (VR Bank Südpfalz, Sitz: Landau)

Ausgabe 232

1 Quartal 2013

Das Vorwort

http://www.womoteam.de/
http://spc.tlienhard.com/

Willkommen zu der Zeitschrift von Usern für User. Wir sind vor allem auf EURE Artikel angewiesen. Ich kann alleine keine (angepeilten) 24-32 Seiten füllen, so gerne ich es auch tun würde. Ehrenwort! Für eingeschickte Artikel gelten folgende Regeln:
Die Artikel müssen sich mit dem Spectrum, ZX81, SAM Coupé, Sprinter 2000 oder nahen Verwandten des Sinclair ZX Spectrum befassen, auch Artikel über passende Hardware und Software sind gerne gesehen.
MAC/PC Software: Nur wenn ausdrücklich direkt im Zusammenhang mit den eingangs erwähnten Computern. Sehr gerne: Crosscompiler, Emulatoren, Game Maker und dergleichen. Auf keinen Fall aber Remakes von Spielen alter Plattformen auf moderner Hardware.

Eigentlich dachte ich, dass ich mit der aktuellen Ausgabe nie anfangen werde. Es ist einfach zu viel zu tun, und es wird nicht weniger. Nach dem Tod meines Vaters will ich einen großen Teil seiner Geräte auf eBay verkaufen. 20 DVD Recorder, Bluray Player, VHS Recorder, dazu viele Zeitschriften und Bücher, seine Spectrum-Sammlung. Ich brauche vermutlich nicht mal die Hälfte davon.
Von dem Geld wird dann ein 3D-Drucker gekauft.
Was ist mit mir? Ich bin in einem AMS-Kurs zwei mal innerhalb einer Minute ohnmächtig geworden. Da programmierte ich ein Spectrum Spiel und hörte nebenbei was der Vortragende so sagt (er hatte übrigens mal den C64 und sein Traumcomputer war der Spectrum), als ich auf einmal Schmerzen spürte. Das nächste woran ich mich erinnern konnte war, dass ein Kollege fragte was mit mir los ist. Ich wollte dann weiter programmieren, fühlte mich aber unglaublich müde und war innerhalb von Sekunden wieder weg.
Das ist nicht witzig! Es bescherte mir eine Woche Krankenstand in dem ich das Clubinfo 231 erstellte.
Ein paar Wochen danach habe ich Wilko Schröter geholfen bei seinem neuerlichen Umzug. Dabei machten wir die Erfahrung dass Umzugsfirmen nicht unbedingt alle vertrauenswürdig sind.

Tatsache ist, dass ich jetzt Angst habe, dass es wieder passieren kann, vielleicht sogar beim Autofahren. Und ich glaube kaum, dass es für mich oder andere Verkehrsteilnehmer lustig wird, falls es wieder passiert. Aus dem Grund habe ich beschlossen dieses Jahr nicht nach Wittenberg zu fahren.

Eigentlich habe ich auch derzeit, so kurz nach dem Tod meines Vaters, auch nicht unbedingt Lust darauf das SPC Info zu machen.

Durch diese Tatsache bedingt, verzögerte sich das SPC-Clubinfo erneut ganz erheblich.

LCD-Leszek Chmielewski

Termine 2013

19. - 21.04.2013: ZX-TEAM Treffen
D-36145, Hofbieber, (Mahlerts)
Die Hardwareprofis für den ZX81 und ZX-Spectrum mit originellen und innovativen Ideen!

27. - 28.04.2013: Vintage Computerfestival
Mehrzweckhalle des ESV, Baumkirchner Straße 57, D-81673 München
Europa (VCFe 14.0) in München

Hauptthema: Lernen
Mehr unter www.vcfe.org/D/

04.05.2013: Die lange Nacht der Computerspiele
D-04289, Leipzig, Karl Liebknecht Str. 145, Beginn 16 Uhr. Ende um 3 Uhr Sonntagmorgens. Im Mittelpunkt steht das Ausprobieren alter wie neuer Spiele. Entwickler und Studenten zeigen ihr Schaffen, Sammler ihre Videospiel-Schätze. Die Roboter-Fußballmannschaft der HTWK stellt sich vor. Eine Vitrinen-Ausstellung zeigt historische Spiele und Hardware .... uvm Weitere Infos unter www.schreibfabrik.de

07. - 08.09.2013: Spectra-Joyce
Wolfhagen
Die Joyce User-AG und der Spectrum Profi Club beschäftigen sich mit CP/M Rechnern (Joyce) und Sinclair Computern

24. - 25. 08.2013: Spectrology
Kulturbund, 06886 Wittenberg, Lutherstr 43 a
Schwerpunkt sind der ZX-Spectrum und ZX81, aber auch andere 8-Bitter Freunde haben dieses Treffen in der Lutherstadt für sich entdeckt!

noch offen: Xcentrix
Seeshaupt, Oberbayern
Das 16. XzentriX Treffen wendet sich an Computerfreunde, die an ein Leben nach der Standardsoftware glauben.

Foreword-English

Welcome to the magazine by users for users. We are primarily dependent YOUR article. I alone can not fill (targeted) 24-32 pages, even I would like to do it. Word of honor! To be sent article, the following rules:

The articles have to deal with the Spectrum, ZX81, SAM Coupe, Sprinter 2000, or close relatives of the Sinclair ZX Spectrum, including articles on appropriate hardware and software are welcome.
MAC / PC software: Only if expressly directly in connection with the above-mentioned computers. I would be very happy about: cross compiler, emulators, Game Maker, and the like. In no case, however, remakes of old games on New Plattforms.

Actually, I thought that I would never start with the current issue. There is just too much to do, and it will be not less. After the death of my father I want to sell a large part of its equipment on eBay. 20 DVD Recorders, Bluray players, VHS recorders, also many magazines and books from his Spectrum Collection. I probably do not even need half of them.

Whats about me now? I've became twice a faint within one minute. Since I programmed a Spectrum game and heard in passing what the lecturer says (He had incidentally the C64 and his dream computer was the Spectrum) when I felt pain all at once. The next thing I could remember was that a colleague asked me what is going on. I then wanted to continue programming, but felt incredibly tired and was gone within seconds.
This is not funny! It gave me a week of sick leave in which I created at the club info nr. 231.

The fact is that now I'm afraid that it may happen again, perhaps even while driving. And I hardly think that it is funny to me or other road users if it happens again. For the reason I have decided to not go to Wittenberg this year.

Actually, I have currently, so soon after the death of my father, not necessarily like to make the SPC Info.
Due to this fact, the SPC Club Info delayed again considerably.

LCD

Neuigkeiten für unseren „Alten“

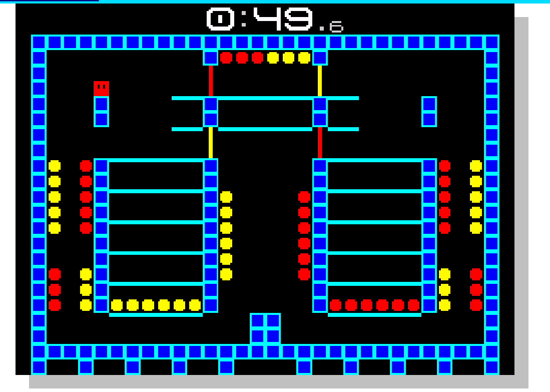
Die Panzerfahrer

http://zx.pk.ru/showthread.php?t=21060

Introspec hat sein Spiel „Tank Battle“ aus dem Jahr 1995 aufpoliert, fehlerhaftes Englisch ausgebessert und ein gepacktes TAP erstellt. Respekt!

Juwelenjäger

http://www.worldofspectrum.org/infoseekid.cgi?id=0028157



Gem Chaser ist die Umsetzung eines xBox 360 Spiels. Man muss Juwelen in selber Farbe wie man selbst ist einsammeln. Man kann seine Farbe beim Durchschreiten bestimmter Barrieren auch ändern.

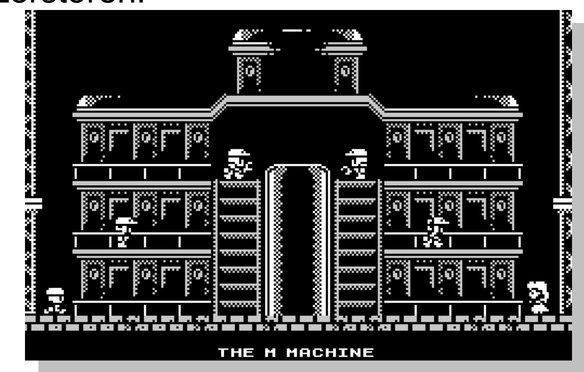
Angriff der bösen Roboterfrau

http://www.worldofspectrum.org/forums/showthread.php?t=43024

Nein, Mojon Twins bringen kein neues Phantomasa heraus, sondern ZXBeccy arbeitet gerade an einer Spieleumsetzung des Klassikers von Fritz Lang: „Metropolis“.



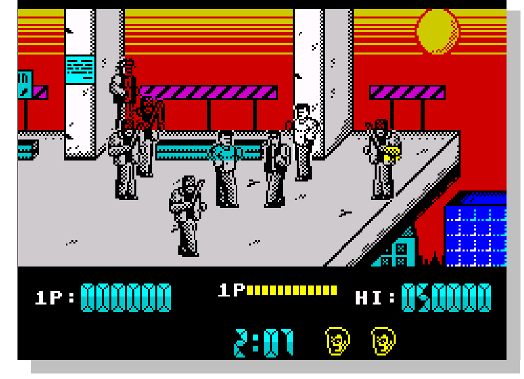
Als die falsche Maria muss man alle Ventile abschalten und die Stadt so zerstören.



Richtig gelesen. Man ist in dem Spiel ein böses Robotermädchen. Das Spiel wird mit AGD geschrieben und hat Monochromgrafik.

Prügelknabe

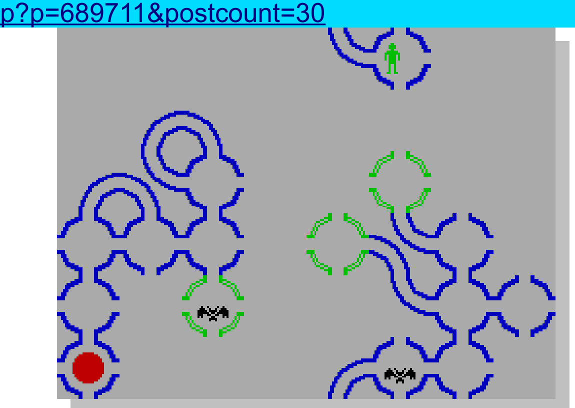
http://www.worldofspectrum.org/forums/showthread.php?t=42945



Ralf hat ein Renegade-Mod erstellt mit veränderter Grafik und etwas modifiziertem Code, der die Spielbarkeit verändert.

Bringt den Wumpus zur Strecke!

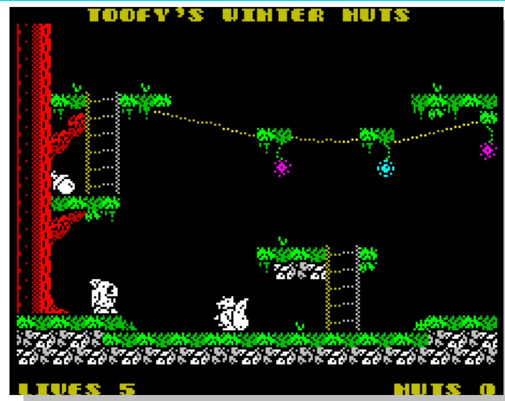
http://www.worldofspectrum.org/infoseekid.cgi?id=0028119
http://www.worldofspectrum.org/forums/showpost.php?p=689711&postcount=30



Die grafische Umsetzung von „Hunt the Wumpus“ für TI99/4 aus dem Jahr 1980 hat mich fasziniert und ich hielt die Umsetzung für den Spectrum für leicht machbar. Lange Rede, kurzer Sinn: Der Klassiker wurde von mir fast 1:1 für den Spectrum umgesetzt, natürlich mit Boriels ZX BASIC Compiler. Und als ob das nicht genug wäre, Dr Beep wurde davon inspiriert und erstellte eine etwas reduzierte Version 1K ZX81 mit Pseudo-Hires-Grafik.

Toofys zweiter Fall

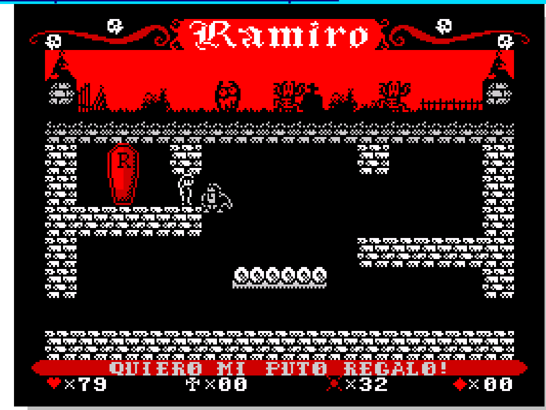
http://www.worldofspectrum.org/infoseekid.cgi?id=0028158



Toofy's winter Nuts ist das Nachfolgeprojekt zu Toofy in fan land. Wie man sieht, hat sich die Grafik stark verbessert.
Auch das Spiel ist besser geworden.

Ramir der Vampir

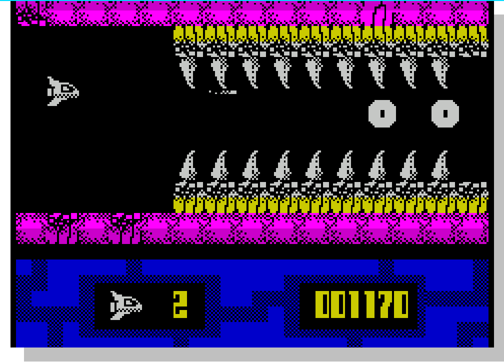
http://www.mojontwins.com/juegos_mojonos/ramiro-el-vampiro-ramire-the-vampire/



Als Vampir in Ramire the Vampire hat man so Schwierigkeiten im (untoten) Leben. Erst einmal reagiert man allergisch auf das Sonnenlicht, und dann stellt sich noch heraus, dass man den Todestag der Tochter vergessen hat, und diese ist natürlich sauer.
In dem 2-teiligen Spiel von Mojon Twins sammelt man wieder Gegenstände und löst Rätsel. Es ist ganz nett geworden und atmosphärisch sehr dicht. Es läuft mit 48 K und der Z88dk Sourcecode ist verfügbar. Es gibt eine spanische und englische Sprachversion.

Moderate Retribution

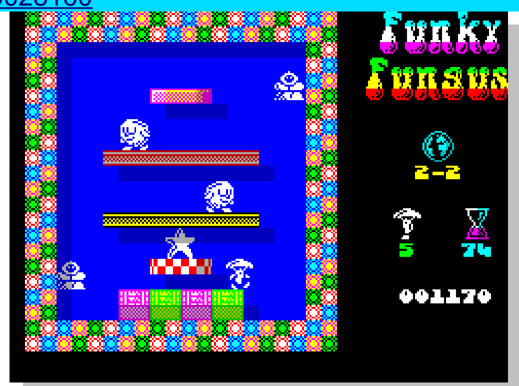
http://www.worldofspectrum.org/forums/showthread.php?t=42100



Moderate Retribution ist ein SEUD-Shoot'em up von R-Tape und gar nicht mal so übel. SEUD scheint mir ein mächtiges Werkzeug zu sein. Ich bin aber eher ein Fan der normalen Programmierung.

Magischer Pilz

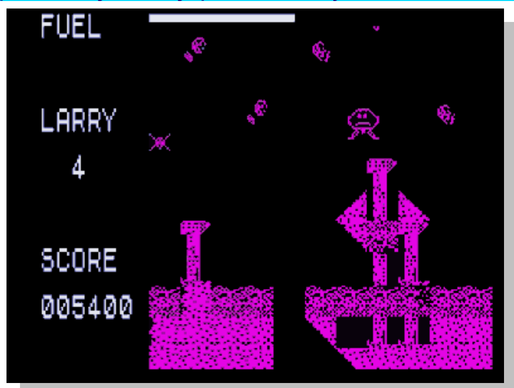
http://www.worldofspectrum.org/infoseekid.cgi?id=0028160



Mit dem Arcade Game Designer 3.4 entstand „Funky Funus“ – ein grafisch aufwendiges und gut spielbares Spiel von Alessandro Grussu aus Italien.
Verfügbar ist das Spiel in den meisten Formaten, inklusive TR-DOS, und diversen Sprachen, sogar Deutsch.

Larry landet lieber leise

http://www.jammajup.co.uk/larry-the-lander.html



Larry the Lander ist eigentlich ein Compobeitrag zu „CompSysSinclairCrap GameCompo“, also im Normalfall nicht berichtenswert. Doch das ist eine Ausnahme weil das Spiel so schlecht nicht ist. Es erinnert an kommerzielle Produkte die so in den 80er Jahren angeboten wurden.
Das Spiel stammt von „Purple Unicorn“ und ist sehr reich an Action und Scrolling. Ach ja. Es wurde mit dem Shoot'em Up Designer entwickelt.

Geadelt

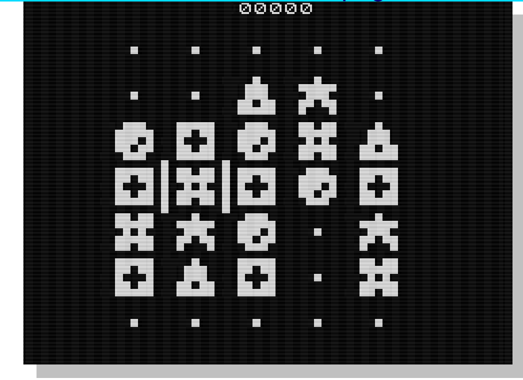
http://www.worldofspectrum.org/infoseekid.cgi?id=0028146



Mit „Sir Vamperi“ hat Carlos David Diaz den Nachfolger zu „Vamperi“ programmiert. Als spanisches Produkt ist das Spiel in spanischer Sprache verfügbar.
Das Siel ist zwar in BASIC und nur als Z80 Snapshot verfügbar, aber keinesfalls schlecht.

Verschoben!

http://www.revival-studios.com/?page=173



Revival Studios hat das Spiel „Shiftet“ für Sinclair ZX 81 mit 16K vorgestellt.
Es ist ein Puzzle-Spiel in dem die Reihen verschoben werden um Kombinationen herzustellen.
Auf Kassette kostet das Spiel € 7,99.

Krusty's Fun House

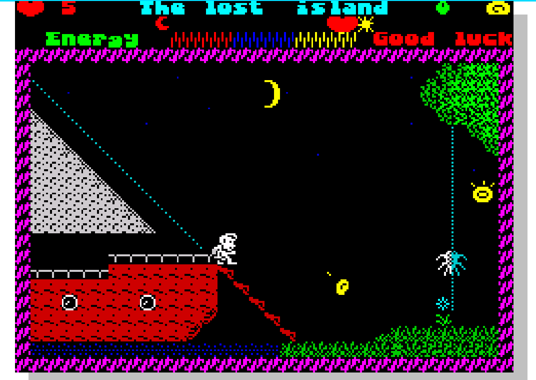
http://www.worldofspectrum.org/forums/showthread.php?t=42992

Sadako hat angekündigt Segas „Krusty's Fun House“ für den Spectrum

umzusetzen. Vielleicht wird es ja dieses mal keine Vapourware weil es mit AGD programmiert wird, obwohl ich meine Hand dafür nicht ins Feuer legen würde dass es fertiggestellt wird.

„Lost“

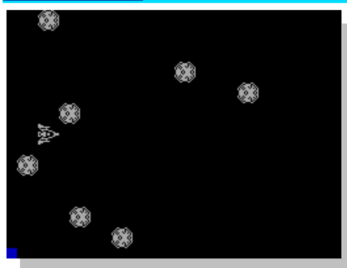
http://www.worldofspectrum.org/forums/showthread.php?t=43154



Mit „The lost island“ ist ein neues Action-Adventure aus Russland erschienen. Ungewöhnlich ist, dass es mit AGD geschrieben wurde. In Russland programmiert man noch selber! Auf jeden Fall sehr interessantes Spiel.

Asteroiden auf 12 Uhr!

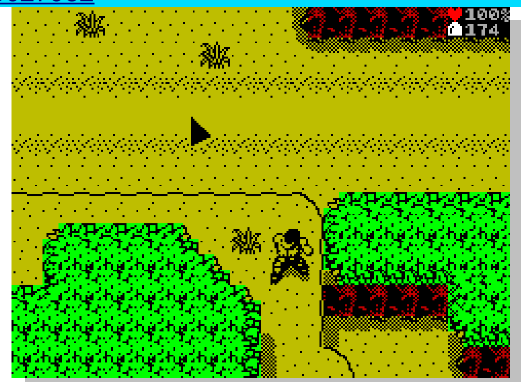
http://www.worldofspectrum.org/infoseekid.cgi?id=0027976



SaNchez hat mit diesem Spiel in 512 Bytes den ersten Platz in Next Castle Party gewonnen. Leider hat das Spiel nichts mit „Asteroids“ zu tun. Man weicht hier den anfliegenden Asteroiden aus.

Survivisection

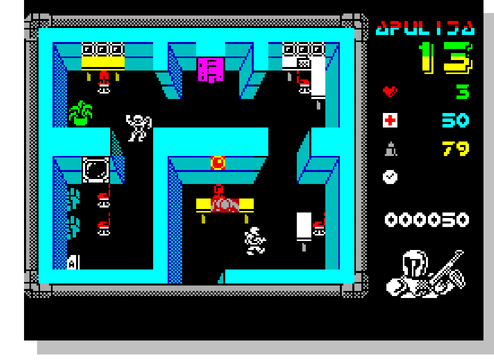
http://www.worldofspectrum.org/infoseekid.cgi?id=0027982



SaNches hat das Chaos Engine-Spiel für den Spectrum zwar fertiggestellt, aber die TAP Version dieses großartigen Spieles, die er vollmundig für September 2012 angekündigt hat, ist noch immer nicht erschienen.
Aber keine Sorge! Hacker sorgen schon dafür, dass auch DivIDE- und Kassetten-Nutzer das Spiel spielen können.
http://velesoft.speccy.cz/other/Survivisection%20(2012)(Sanchez)_fixed_mouse.zip

Apulija 13

http://www.worldofspectrum.org/infoseekid.cgi?id=0027997



Alessandro Grussu aus Italien hat mit dem Arcade Game Designer dieses nette Spiel erstellt, und witzigerweise gibt es das nicht nur in Englisch und Italienisch, sondern auch in Spanisch und Deutsch. Und grafikmäßig weiß es auch zu gefallen.

Wer reitet da durch Sand & Wind?

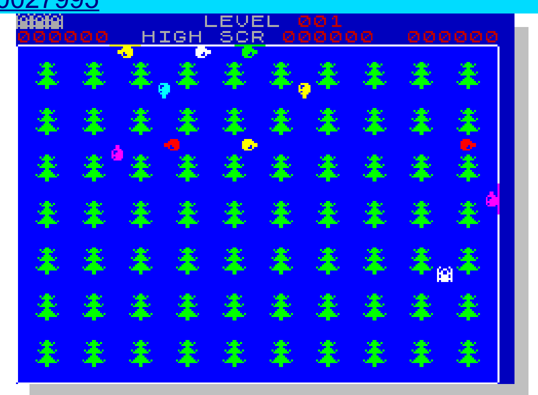
http://www.worldofspectrum.org/infoseekid.cgi?id=0027998



J.B.G.V. ist genauso wie ich ein Fan des ZXBC. „3 Reyes Magos“ wurde daher mit ZXBC programmiert. Man spielt einen der heiligen 3 Könige auf einem Dromedar und weicht Meteoren aus. Eine Anspielung auf „Harrier Attack“ kann man im Screenshot leicht erkennen.

Angriff des Weihnachtsschmucks

http://www.worldofspectrum.org/infoseekid.cgi?id=0027995

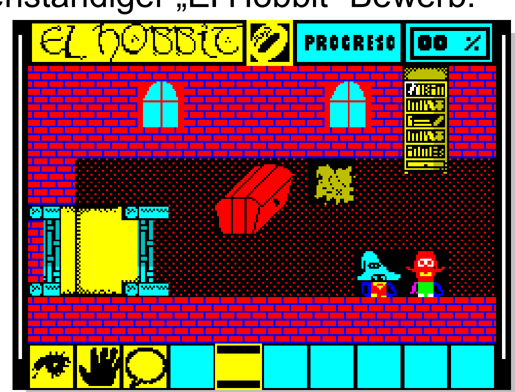


Bauble Blast ist Mole Rat mit veränderter Thematik. Dave Hughes hat Recycling betrieben um sehr schnell ein Spiel mit Weihnachtsthematik zu schreiben. Schlecht ist es aber deswegen nicht.

Die Hobbit-Verschwörung

http://www.worldofspectrum.org/infoseekid.cgi?id=0027992

„El Hobbit“ ist ein Teilnehmer im Bytemaniacos „El Hobbit“ Wettbewerb. Es ist ein grafisch schönes Adventure, wenn auch schwer zu verstehen. Verantwortlich zeichnet sich Alejandro Valero aus Spanien. Derzeitiges WOS-Rating von 6,44 bescheinigt dem Spiel aber gewisse spielerische Qualitäten, die ich ihm keinesfalls absprechen möchte. Leider habe ich in letzter Ausgabe einen Fehler gemacht. Es gab nämlich nicht einen, sondern zwei Wettbewerbe: Der angesprochene BASIC RPG Bewerb, und ein eigenständiger „El Hobbit“ Bewerb.



Glückliche Chips

http://www.chip-kiosk.de/c64



Chip hat eine „Happy Computer

Sonderausgabe“ angekündigt, mit Specials zu C64. Zitat: Die Welt des C64, von A wie Atari bis Z wie ZX Spectrum.
Gedruckte Ausgabe kostet € 9,90 + Versand, eine PDF-Download Ausgabe gibt es schon um lächerliche € 6,50.
Verfügbar ab August oder so.

Ärger in Maldita Castilla

http://rincondelspectrum.blogspot.co.at/2012/12/algo-se-cuece-en-la-maldita-castilla.html

Dieses Spiel schaut nur aus wie Ghosts'n'Goblins, es ist aber ein neues Projekt, an dem der Grafiker von Retroworks beteiligt ist.
Hoffen wir, dass schon bald genaueres bekannt wird.

Zwei neue Werke unterwegs

http://www.jammajup.co.uk/purple-unicorn-software.html



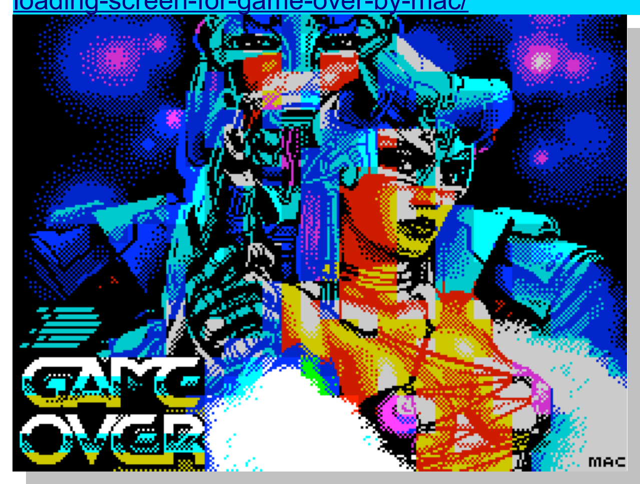


Purple Unicorn Software arbeitet an zwei neuen Titeln:
Barry Bumble und Terry the Turtle. Das erste Spiel ist ein Shoot'em up (SEUD), während das zweite ein Jet Set Willy Klon ist. Nuff said?
Nachdem die Spiele immer noch nicht fertig sind, hat Purple Unicorn Software aber noch weitere Spiele angekündigt:
Jammajupbooble (Bubble Bobble):
Pestilence (P.A.W.)
Harry the Hedghog (AGD)
Louise the Ladybird (?)
Percy the Pigeon (Noooooo!!!)
Pipiiaraddon (?)



Bilder-Kriege

http://speccy-live.untergrund.net/2013/04/new-loading-screen-for-game-over-by-mac/



Wer öfters auf World of Spectrum Website unterwegs ist, hat sicher

bereits gemerkt, dass einige Künstler die Titelbilder bei Spielen neu zeichnen, wo sie der Meinung sind, dass diese schlecht wären. Teilweise entstehen da unglaubliche Meisterwerke. Ich habe mal „Game Over“ als Beispiel herausgepickt. Zorro wurde mit gleich zwei Screens bedacht.

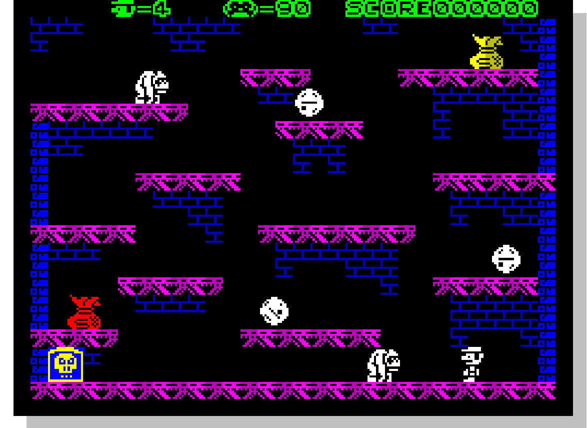
Kompressionsmeister

http://www.worldofspectrum.org/infoseekid.cgi?id=0027996

Einar Saukas hat mit ZX7 einen neuen optimierten PC-basierenden Kompressor entwickelt. Dekompressionsroutinen für ZXBC und Z88dk hat er natürlich gleich mitgeliefert.

Monkey Fong

http://www.worldofspectrum.org/forums/showthread.php?p=680603#post680603



„Donkey Kong Reloaded“, das AGD Spiel, ist nun fertig und darf gespielt werden. Die Grafik ist sehr gut geraten.

Boriels ZX BASIC Compiler

http://www.boriel.com/wiki/en/index.php/ZX_BASIC:Archive#Latest_Development_Version

Boriels ZXBC wurde auf Version 1.3.0s1022 upgedated. Viele Bugs wurden beseitigt. Es wird gerade an Refactoring für die Version 2 gearbeitet. Es entstehen immer mehr Spiele mit ZXBC und dieser Trend scheint weiterhin anzuhalten.

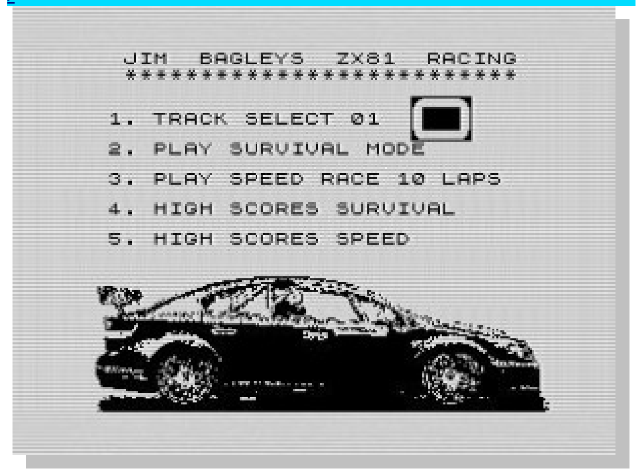
SCR Plus ULA

http://speccy-live.untergrund.net/2013/02/scrplus-v0-21-released/

SCR Plus ist ein Konverter zu ULA+ Bildern. Version 0.21 enthält einige Verbesserungen.

Dschim Baglys ZX81 Rejcing

http://mattyongames.wordpress.com/2013/02/14/793/



„Jim Bagley's ZX81 Racing“ ist ein neues Action-Spiel für ZX81 16K und erinnert etwas an Supersprint.
Auf jeden Fall toll was da für den ZX81 programmiert wurde.

Zeddy Puyo

http://speccy-live.untergrund.net/2013/02/brand-new-game-zx-puyo/



Und wieder ein neues ZX81 Spiel, und zwar ein bekannter Puzzler im Char Modus. In diesem Modus ist der Aufbau des Bildschirms sehr schnell, was der Spielbarkeit zuträglich ist.

Neues Spiel, neues Glück

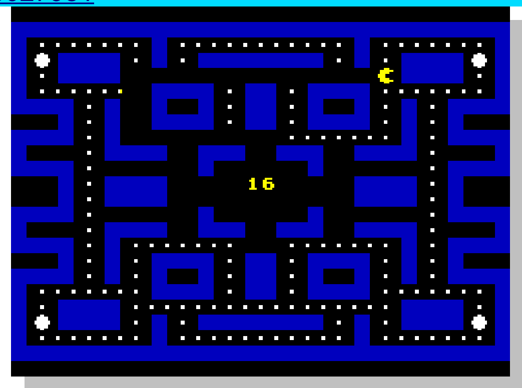
http://foro.speccy.org/viewtopic.php?f=6&t=3096&sid=92c05ffade33cdba5cd69b295354356f

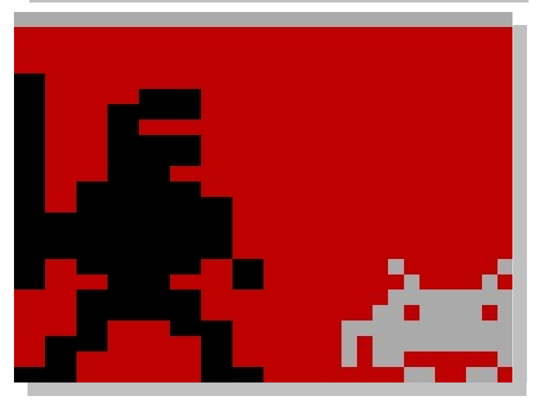


„Escape from the holy Tower“ scheint einen zweiten Teil zu bekommen. Geplant sind 100 Räume, Shops mit Gegenständen und so weiter.

So wenige Bytes...

http://www.worldofspectrum.org/infoseekid.cgi?id=0027978
http://www.worldofspectrum.org/infoseekid.cgi?id=0027981

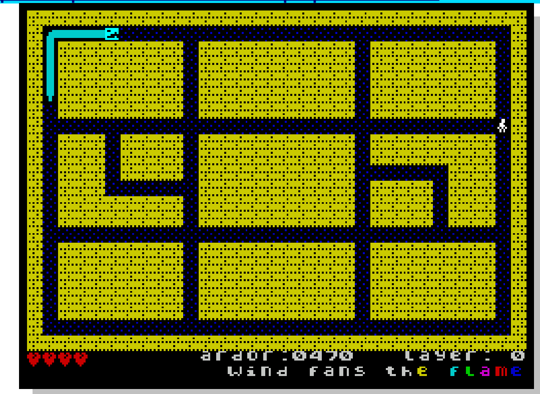




Den dritten Platz beim Next Castle Party 2012 512Bytes Game belegte „Pacmam“ von thesuper, welches mit „Bye, V3nom!“ konkurrierte. Dass 512 Bytes zu wenig sind für opulente Multicolor-Schlachten, sollte jedem klar sein. Trotzdem zaubern die Coder erstaunliches auf dem Computer.

Renn, Forrest Gamp!

http://zx.pk.ru/showthread.php?t=20223



Jerri hat eine Konversion eines alten Spieles für russische Computer am Spectrum vorgestellt. Das Spiel „Run Away“ ist spielbar aber noch nicht fertig.

Olisa

http://www.worldofspectrum.org/infoseekid.cgi?id=0027921

```
TIME - see session time and who is here
GO - listen w/o participation
DELAY - request more session time
LEAD - lead Olisa to Forum
HELP - see this list of commands
EXIT - terminate program


Pandora, 2189
Secret animal psychology research facility

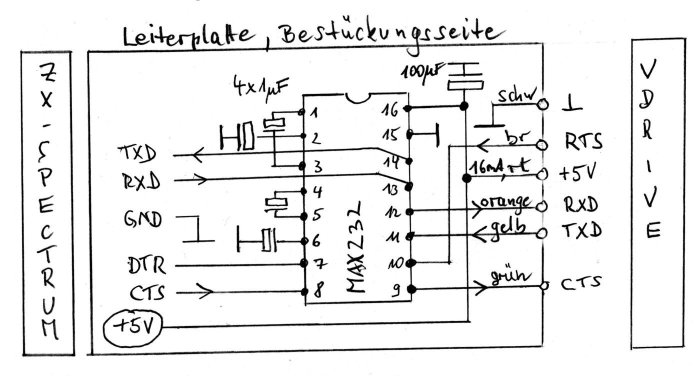
WARNING: perimeter violation
COCO: base to THIRD, details please
THIRD: the violator is aboriginal, behaviour not aggressive
NIM CHIMSKY: it's OLISA, a girl from the village
NIM: must've gone astray in the forest
COCO: shall we evacuate the perimeter?
NIM: too dangerous, she may encounter the Predator
NIM: if she's lost, the villagers will scan the entire forest
NIM: we'll all have to retreat then
COCO: damn secrecy! can we not help her?
NIM: wait, I've got an idea
```

Dimitry Danner hat mit „Olisa“ ein Textadventure in Englisch und Russisch programmiert. Ob das wohl etwas für unsere Adventure-Ecke ist?

Mehr News gibt es nächstes Mal.

USB-STICK am ZX-Spectrum mit IF1, Teil 2

Zu meinem Beitrag im Clubheft 229 zum Thema USB-Stick am ZX-Spectrum erreichten mich verschiedene Anfragen zu weiteren Details. Ich freue mich über das Interesse und möchte gern einige Einzelheiten darlegen und über neue Erfahrungen berichten.



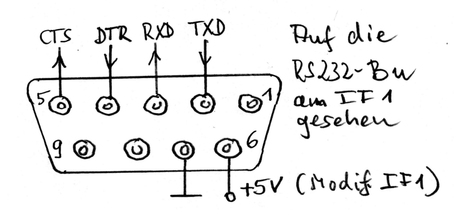


Bild 1

Die erste Gruppe der Fragen bezieht sich auf die technischen Einzelheiten und einen Schaltplan (s. Bild 1). Kernstück ist der IC MAX232, der die RS232-Pegel von –12V..+12V auf TTL-Pegel 0V..5V umsetzt und die Signale gleichzeitig invertiert. Erinnern wir uns: Im RS232-Standard hat das logische HIGH den Namen MARK und den Pegel –12V, das logische LOW heißt SPACE und gehört zu +12V. Das VDRIVE braucht aber HIGH=+5V und LOW= 0V. Diese Umsetzung vollzieht in beiden Richtungen der MAX232. Die 4 Kapazitäten von 1uF sind zur internen Spannungsvervielfachung unverzichtbar. Im IF1 muss intern eine Leitung gezogen werden, die+5V über die RS232Buchse nach außen führt. An die Ausgänge wird dann über die mitgelieferte Steckerleiste das VDRIVE angeschlossen (s. Bilder 2 und 3). An der werksseitigen Jumpereinstellung wurde nichts geändert.

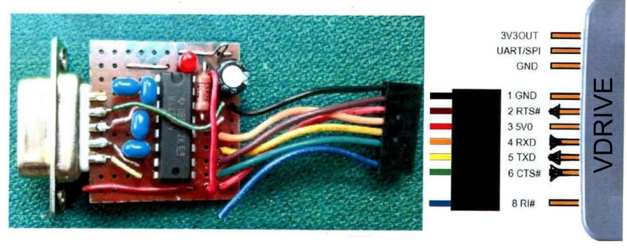


Bild 2

Erste Versuche erfolgten mit dem in Heft 229 beschriebenen BASIC-Programm, inzwischen ist aber ein MC-Programm vorhanden, das sich bestens bewährt hat. Ich will einige Grundprinzipien dieses ca. 700 Bytes langen Programms erläutern. (In einer Sparversion kommt man sogar mit ca. 400 Bytes aus). Die Kernidee besteht darin, die SAVE/LOAD-Befehle und weitere nützliche Kommandos für die Steuerung des VDRIVEs als NEUE BASIC-Befehle einzuführen. Das geschieht durch Änderung der Systemvariablen VECTOR und ist z.B. in [LOGAN], Seite 124ff beschrieben. Auf diese Weise stehen die folgenden Befehle zur Verfügung:

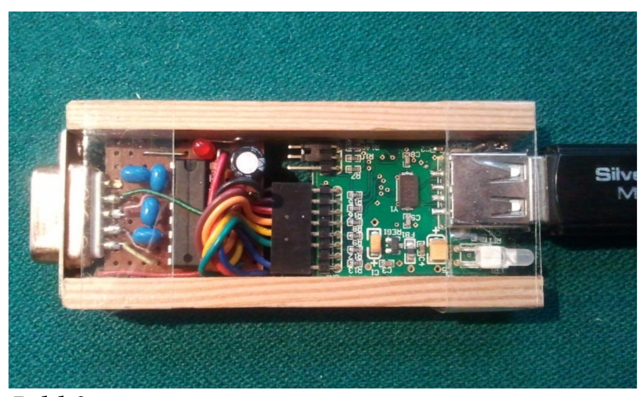


Bild 3

!D=directory, !R=read (bis PROMPT

>),

!L"filename"=LOAD, !S"filename"= SAVE,

!E"filname"=Erase.

Dabei sind durch massive Verwendung vorhandener Routinen aus dem Haupt-ROM und aus dem IF1-ROM problemlos auch die Erweiterungen

!S"filename"SCREEN$,

!S"filename"CODE nn,mm

und die entsprechenden Ladebefehle möglich. Als "filename" sind auch Stringvariable, und für nn, mm auch numerische Ausdrücke zugelassen. Ein Beispiel ist das folgende Programm, das wechselseitig zwei Screenbilder lädt:

10 !L"BILD1"SCREEN$: !R: PAUSE 100

20 !L"BILD2"SCREEN$: !R: PAUSE 100

30 GO TO 10

Bei der Auswahl der Befehle habe ich mich auf die am häufigsten verwendeten beschränkt. Will man aber weitere Befehle an das VDRIVE schicken, z. B. Change Directory, so ist das einfach auf folgende Weise möglich:

OPEN #4,"B": PRINT #4;"CD NeuesDir": !R

Auf diese Weise steht letztendlich der gesamte Befehlsvorrat des VDRIVE zur Verfügung.

Eine praktische Erfahrung, über die mir im Nachhinein auch T0m berichtete: Große USB-Sticks (ab 1GB) machen Schwierigkeiten: Bei SAVE/LOAD von Dateien >2kB gibt es häufig Fehler, wohl durch Timingprobleme verursacht. Die besten Erfahrungen machte Tom mit Sticks von 32MB. Versuchsweise habe ich statt eines USB-Sticks einen CardReader mit eingelegter CF-Card von 1GB angeschlossen: Alle Ladeprobleme waren wie weggeblasen. Seither läuft damit alles problemlos, große Dateien sind kein Problem mehr, und z.B. auch die beliebten MANIC MINER und AntAttack laden reibungslos. Zu den Ladezeiten: Da alles über die Befehle LOAD*"B" und SAVE*"B" geht, kommt man auf ca. 9000Baud=1kB/sec. Das ist nicht übermäßig schnell, aber der große Komfort der CF-Card macht das wett.

Das MC-Program selbst habe ich in einem umprogrammierten Druckerinterface LPRINTIII zusammen mit einem neuen, kurzen Druckertreiber untergebracht. Von dort wird es mit COPY in den RAM von #FA00-#FCFF geladen.

Danksagung: Mein großer Dank geht an Roelof Koning, mit dem ich hierzu sehr ausführliche Diskussionen hatte und ohne dessen Hilfe das Programm nicht so geworden wäre.

Weiteren Nachfragen oder Anregungen erwünscht!

Heinz Junek

Literatur:

[LOGAN] Ian Logan, Das Microdrive Universum, Hüber Verlag 1984.

Debunking Commodore 64 speed myth

http://www.alfonsomartone.itb.it/aunlzr.html

Yes, I found a **funny** article on a Commodore 64 fan site, claiming to show that the C64 processor (the MOS 6510) had at least something computationally faster than the Z80A of the Zx Spectrum.
That was a false statement. In fact, the 3.5 MHz Z80A of the Sinclair Zx Spectrum kicks the 0.98 MHz MOS 6510 in the ass (hard!).

That article ("ffd2/fridge/speccy/score") contained, as in the Most Standard Commodorian-Fan Speech, factual inaccuracies, fake assumptions, unproven assertions, stupid slogans, ...

Below, I will quote everything he says, adding my comments. Since we are showing Z80-class and 6502-class features, "Z80" and "Z80A" will be considered synonyms, and also "6502" and "6510".

The background:

The Spectrum consists of a Z80 running at 3.54 MHz, a 256x192 bitmapped display, a simple sound chip, and 48k. The C64 consists of a 6510 running at 1MHz, a sophisticated sound chip, 64k, and a sophisticated video chip, offering 16-colors, text modes, custom character modes, 320x200 bitmap modes, 160x200 multicolor bitmap modes, sprites, raster interrupts, blah blah blah and so on.

First, let me say that European version of the Commodore 64 was less than 1 MHz. It was 0.98 MHz, yes, zero-dot-nine-eight, yes, 982 kHz. Except the Commodore 64, I've never seen a computer running at less than one megahertz. The Spectrum Z80A "runs" like a cheetah at 3.5MHz while the Commodore 64 "runs" --oops, no, "walks" like a turtle at 0.98MHz. Yes, **megahertz matters**.

Then, let me say that if you want to compare raw processor speed in performing particular tasks, you should save us your "sophisticated chips" sermon (a Commodorian fan standard): these comparisons are performed on "bare microprocessor power", between the 3.5 MHz Zilog Z80A of the Zx Spectrum and the 0.982 MHz MOS 6502 of the Commodore 64.

Note: I will assume the Zx Spectrum running at 3.5 MHz, not 3.54 (the "3.54" thing seems to be Spectrum 128 related).

While it is clear that the 64 will outperform the Spectrum on tasks related to text, sprites, etc. it was not clear what sort of performance ratio existed for straight calculations and applications involving bitmapped graphics: for example, 3D graphics programs.

Again the C64 Sermon, how boring. But -well- let's be more precise.

"Text": while in **text-mode**, the Commodore 64 needs to write one byte to draw a character, and another byte to set up its color. That is, two bytes, plus time required to calculate their respective addresses. The Zx Spectrum does not have a text mode.

Zx Spectrum only has a graphics mode (a full bitmap always available at a fixed address, not requiring any initialization/setup). It requires eight bytes to be written into display memory to form an 8×8 character cell, plus an attribute byte to set up its color. That is, nine bytes, plus time required to calculate cell address and attribute address (due to its peculiar screen mapping, going through the 8 scanlines to write a character requires only a simple 8-bit register increment, which is as fast as a "NOP" instruction).

Thus, the C64 "outperforms" only when in text-mode compared to graphics bitmap mode. That is, "outperforms" only when you compare bananas to apples...

Sprites: yes, you "outperform" on sprites, yes, Zx Spectrum does not have sprites. Commodore 64 is a "professional computer" (as of its label below the keyboard) which has "sprites" (the standard videogames feature). "Professional" with "sprites", something like putting Coke into Spumante wine. Great. I always told you that Commodorian fan sermons make me either laugh or bore.

"Etc"... er... what's "etc"? Where are the other "outperformings"?

A number of people claimed that the Spectrum would significantly outperform the C64 on general calculations and 3D graphics programs; these claims were based primarily on the 3.54:1 MHz ratio and personal bias, but were certainly claims worth investigating.

Investigations are welcome, but the 3.5MHz to 982kHz ratio *is always significant. It's not a 5-10%, it's three times and a half (3.564 more precise figure); if you try to erase this* **fact**, *then the "personal bias" you're blaming is... yours!*

I began looking for insight into three main questions:

1. How do the Z80 and 6502 differ, and what is the Z80 approach to solving problems/how does it differ from the 6502 approach?

2. How would the Spectrum perform on the kinds of programs that I write?

3. Is there any merit to the earlier claims about the clear superiority

of the Spectrum a) in general and b) specifically for 3D graphics?

Hmmm... the kinds of programs that "you" write?

Some extra small questions: do you write Z80 assembler software as well? Did you ever use the second set of Z80 registers? Do you know its "repeating" instructions like "LDIR", "CPDR", "INIR" and so on? Did you ever use their non-repeating equivalent ("LDI", "CPI", "OTI" and so on) to get multiple things in one instruction? Did you ever test the so-called "undocumented" Z80 instructions? (in spite of their name, they are largely used and known because often help to save time while programming and execution size and speed).

Looking at the results below, I guess not. It seems that, except some notable cases (like the "LDIR" of note 1) you are using the Z80A as if it was an 8080 or a 6502...

Some technical discussion eventually ensued, and a number of code snippets were compared. Cycles were counted, opinions were declared, but through it all a few things became apparent, and new insights were gained.

Alas, we don't find "where" you counted the cycles. In the article you present a list of figures without any line of code, plus a dead link to a non-existing web site. And "new insights" seems just another name for "Commodorian bias".

General conclusions:

1. The people who were loudly extolling the virtues of the Spectrum over the 64 not only did not understand the broad issues (i.e. have any practical 6502 or C64 programming experience to compare with), they did not even understand the specific issues (e.g. practical experience in doing 3D graphics, or drawing lines).

This statement applies to Commodorian people as well. There are not much people out there programming on both Zx Spectrum and Commodore 64.

*But, as we will see later, the statement applies **only** to Commodorian fans.*

2. The typical cycle ratios are around 3:1.

Seven programs have been considered: slow multiply, block mem transfer, substring search, three line routines, and the fast multiply. The slow multiply runs at 2:1. The non-LDIR memory copy runs at 3:1. The substring search typically runs at 3:1. The line routine runs at

3:1, with unrolling bringing it down to 2.7:1. In practical use (e.g. a matrix multiply) the fast multiply runs at > 3:1.

You should say what you mean for "runs at 2:1". Is this a cycle count? Then the times must be multiplied for the clock ratio, because Zx Spectrum has 3.5 million "clockcycles" in a second, while the Commodore 64 has 981 thousand "clockcycles" every second.

Or, was it a real figure? Does it mean that Zx Spectrum "runs at 2:1" (two times faster than Commodore 64)? I guess this is exact. But... why did you criticize people "extolling the virtues of the Spectrum"? Why did you forget the "virtues" that you were going to demonstrate? Another Commodorian Blah-Blah-Blah, eh?

From this I conclude that typical cycle ratios will be 3:1, in particular for the kinds of programs I write. Obviously, some algorithms will do better, and others will do worse.

Then let's see where "better" and where "worse", and let's actually verify your "3:1 ratio" assertion...

Conclusions to my specific questions:

1. The Z80 is based around its registers. Algorithms which fit entirely within the registers do very well, especially for 16-bit applications.
Memory access is generally done indirectly via (HL), which tends to favor sequential memory access. The stack pointer is 16-bits and may be used for useful things. Branching and jumping are generally slow, as are direct memory address and absolute numbers, and indexing is fairly dissimilar. The Z80 has a number of specialized instructions which are used in a variety of tasks, and lacks the range of addressing modes offered by the 65xx.

First lie: *"branching and jumping is slow". Let's evaluate "BNE" (branch when not equal) on 6502, and the Z80 equivalent "JP NZ". BNE requires 2 clockcycles if not jumping, or 3 clockcycles when jumping; JP NZ requires 4 clockcycles if not jumping, or 10 clockcycles if jumping. The Spectrum is faster than the C64. Demonstration:*

- *JP NZ: 4 and 10 clockcycles at 3.5MHz, that is 1.14ns and 2.85ns*
- *BNE: 2 and 3 clockcycles at 982kHz, that is 2.03ns and 3.05ns*
- *result: branch/jump is slow on the C64; in fact Zx Spectrum shows to be 7% to 78% faster than Commodore 64.*

Oh, and I did not mention the performance hit when the "BNE" jump crosses a 256-byte page boundary, requiring an extra clock cycle... the "BNE" works well only inside a 256-bytes page! Commodorian programmers will have to "encapsulate", as far as possible, their assembler programs in "chunks" of 256 bytes!

Second lie: *"direct memory access is slow". Let's evaluate "INC $nnnn" (increment byte at location nnnn, "direct memory access" of the 6502) with the two Z80 instructions needed to simulate it, that is "LD HL,$nnnn ; INC (HL)": the 6502 "direct" requires only 6 clockcycles, while the Z80's "LD HL,$nnnn" requires 10 clockcycles and the subsequent "INC (HL)" (increment the byte pointed by HL) requires 11 clockcycles, for a total of 21 clockcycles of the Z80:*

- *6502: "INC $nnnn": 6 clockcycles at 982kHz, that is 6.11ns*
- *Z80: "LD" and subsequent "INC (HL)": 21 clockcycles at 3.5MHz, that is 6.00ns*
- *result: Zx Spectrum is still sligthly faster (+1.18%) than Commodore 64. Thus, the slow one is the C64.*

But the first "LD HL" can sometimes be saved (it's only an initialization); if you happen to need to increment two times the same variable, you have:

- *6502: "INC" + "INC" = 12 clockcycles at 982kHz, that is 12.22ns*
- *Z80: "LD" and two subsequent "INC" = 10+11+11 = 32 clockcycles at 3.5MHz, that is 9.14ns*
- *result: Zx Spectrum is 33.6% faster than the slowcoach Commodore 64.*

Commodorian bias: "Z80 lacks addressing modes". But, as shown in the "second lie" case, the Z80 can simulate missing modes and still win the match.

Useful hint: in the famous Z80 "undocumented" opcodes there is quite a number of operations (like "rotating a memory value", better than the 6510) which will save code size and execution time. I think Commodorians already used all of their "powerful" "undocumented" codes as well.

The 65xx is based around fast access to memory, in particular zero page, and its index registers. Algorithms which involve scattered memory accesses do very well, as do programs which make heavy use of branching and subroutines. The ability to add, compare, etc. directly from memory (ADC $C002 : CMP $D020) means that algorithms involving large amounts of variables, tables, pointers, etc. will perform much better on the

65xx. Absolute operations (ADC #$21) are significantly faster (2 cycles on 6510 vs. 7 on Z80). Algorithms involving relatively few variables bog down in comparison.

What does "fast access" mean? Perhaps, other manufacturers like Zilog, build their processor in order to have a "slow access" only? Let's wipe out the Commodorian Bias and let's go on.

Commodorian Bias: "zero page" is the first 256-bytes page. Yes, 256 bytes only. And yes, there are registers and stack and other "untouchable" (or "beware!") data. How many variables and buffers can you store there? The main advantage of "page zero" instructions is not speed, but a memory saving: a "zero page" access saves one byte of code because the address size is only 8 (instead of 16) bits. "Zero page" faster access will save a few clock cycles - while making Commodorian assembler programmers to all-night fights against that ridicolously small area where those single-cycle-and-single-byte-save instructions can be performed. How about a 30+ kilobytes assembly code having to always to rely on a 256-byte page?

Bias Again: "scattered memory accesses do very well". It seems that Zx Spectrum programmers work on clean, ordered software source, while Commodorians work only dirty, scattered, messy, spaghetti-like software source...! Planning a software implies choosing "scattered" and "non-scattered" models. And having a bunch of registers (like those of Z80, absent in 6502) means saving lots of intermediate operations in both cases.

***Third lie:** "heavy use of branching and subroutines". Where do they perform better? We've already seen the "JP NZ" example, let's see the "RET" example now. The "return from sub-routine" instruction of the 6502 is called "RTS" and requires 6 clockcycles on the 6502, while the Z80 equivalent is called "RET" and requires 10 clockcycles:*

- *6502: RTS requires 6 clockcycles at 982kHz, that is 6.11ns*
- *Z80: RET requires 10 clockcycles at 3.5MHz, that is 2.857ns*
- *result: Zx Spectrum requires 53.2% less time than Commodore 64*

***Fourth lie:** "absolute operations... significantly faster". Huh? What does it mean? Let's see an example. Adding the immediate value $21 to the accumulator: "ADC #$21" on the 6502 and "ADD A, $21" on the Z80. On the 6502 it requires 2 clockcycles, on the Z80 it requires 7 clockcycles. But when we do a little calculation, we see that:*

- *Z80: 7 clockcycles at 3.5MHz, that is 2.00ns*
- *6502: 2 clockcycles at 982kHz, that is 2.0366ns*

- *result: Zx Spectrum is 1.183% faster than the "significantly faster" Commodore 64.*

Well... ehm... uh-oh... **When Commodore 64 fans say their machine is "significantly faster", *then it actually means that the Zx Spectrum outperforms the C64.*** And yes, all of my friends "knew" that their C64 was "powerful" and "fast"... har! har! har!

Commodorian Bias: "large amount of variables... bog down...": these claims do not prove anything. To gain some -and disputable- advantage, you "should" have a large amount of messy scattered data, try always to access all of them at once, and work out in a manner that saving data to registers will not help you. Hey, it seems the Art of Heavily Acrobatic Programming! And it also underestimates the Z80 registers, which purpose is not the trivial "store them some data to save some memory access".

```
Thus, Z80 algorithms try to fit all the variables into registers, reduce memory access to sequential or page-aligned accesses (so that HL may be used), try to avoid branching/decision making, and try to use specialized instructions like DJNZ.
6510 algorithms make heavy use of the index registers, zero-page
and absolute operations, don't mind lots of branching/decision making, and try to avoid too many operations on variables.
```

Commodorian Bias again and again *(but much of it has been already debunked above). Well, the above paragraph could be called "Commodorians do not know how to write software". They think that register allocation is done once per software, and never changes at runtime. They think that branching and decision making is faster on the "less than one megahertz" processor, even after comparing actual timings. They think they can "scatter" messy variables everywhere and operate on them anytime. They think processor registers are useless. They think, they think, they think, they think...* ***they stink****.*

```
2. For things like 3D graphics, the Spectrum probably has a 10%-20% speed advantage. While nontrivial, it is by no means decisive. For programs involving any text, sprites, etc. the Spectrum will clearly suffer.
```

Wonderful. After a bunch of false or biased statements, the Commodorian Guy still shows numbers. This time he admits that the Zx Spectrum "probably" ("probably?") has a 10-20% speed advantage (is it "10%" or is it "20%"?); better, a ***"nontrivial" advantage of the Zx Spectrum****. Good. This is one of the unique times we see a Commodore fan admitting that the* **Zx Spectrum is "nontrivially" better than the**

Commodore 64. *But even admitting a "probably 20%" better speed, the Commodorian guy immediately says "by no means decisive". Amazing. I couldn't ask more to demonstrate better the Typical Commodorian Bias.*

You now understand what Commodore fans are. "6502 is better, 6502 is faster, 6502 allows memory access, 6502... 6502... yadda-yadda-yadda... the Zx Spectrum is probably nontrivially faster, but by no means decisive... but we Commodorians have sprites, and Sinclairists don't have sprites! Yay! We're better! You Spectrumists suffer! Yay!"

```
3. You gotta be kidding. As
if. :) Finally, the numbers.
All code may be found at
http://stratus.esam.nwu.edu/
~judd/fridge/
```

Dead link.

```
Shootout at the 0K Corral
-------------------------
(Cycles)
Z80/Spectrum    C64    ratio
------------    ---    -----
8x8->16 shift&add multiply
357/385 160/216/248 1.6-2.2
```

This is indeed the interesting part. They claim to have programmed a number of routines in both Z80A assembler (Spectrum column) and 6510 assembler (C64 column) to see the "ratio" of their respective clock cycles count. We already saw that Spectrum processor runs on a 3.5MHz processor, that is 3.56 times faster than the Commodore one. That is, a routine requiring 3500 clock cycles on the Zx Spectrum will complete in one millisecond; a routine requiring only 982 clock cycles on the Commodore 64 will complete in the same time - one millisecond (yes, megahertz difference matters. To get "head-to-head", the "ratio" should be 3.56).

The "ratio" figure thus says that the Zx Spectrum is either "faster" (when ratio is less than 3.56) or "much faster" (when ratio is "much less" than 3.56; for example, if ratio is 1.78 then it means that the Spectrum is twice as fast than the Commodore 64.

```
Block mem copy 39*x 13 3
21*x [1] 1.62

Substring search [2] [3]
init: 29 4
successful compare: 57 19 3
advance next substr: 46+21*x
11+9*x [4]
advance and loop: 61 15 4.07
compare last char: 40 18
2.22

Line routine [5] 73/111 [6]
30/33 2.92 [7]
29/37 [8]
24/72 (21/68) 6/30 (5/28)
2.7 (2.7)
49/71 21/25 2.6

Fast multiply [9] 100
(+7/3/27) 43/25 [10] 2.3-
2.6/4.2-4.4

Sprites
```

String print None offered [11]

*This is kinda weird to explain. Currently we miss the assembly source for the Z80A routines; the Commodorian guy who presented the above table did surely well on the 6510 side, but... did he wrote optimal Z80 code as well? Or did he rely on some sub-optimal code? (like the one in the teaching books). I think he didn't write optimal code: in the first test he shows a "block mem copy" figure of "39*x", and then says that the "LDIR" version (the most common operation on large block copying on Z80 processors) requires only "21*x". Thus the "39*x" figure comes from a largely sub-optimal example that has been placed there only to show some figure near the 3.56 "head-to-head" value...*

The C64 has better "ratio" values only in two cases (fast multiply and a weird-named "advance and loop" with a funny clockcycle count; more on later).

In all other cases, Spectrum shows the expected figures: 2.2-2.3 means that the Z80A of the Spectrum performs at 155%-165% the Commodore 64 "fast" speed; the almost missing "LDIR" test performs at 225% the C64 "fast memory access" speed...

Notes:
1. Using LDIR on Z80.

2. Given a list of null-terminated strings, find a particular string. The substring search involves four main processes: successful character compare, compare of last character, advance to next substring on mismatch, and advance pointer and loop for next string. Init refers to initial setup (trivial). "Advance to next substring" includes unsuccessful character compare. "Advance and loop" counts cycles up to normal compare loop.

3. The C64 version is 28 bytes. A 64 program would change this problem slightly to improve performance (strings terminated with inverted dextral character instead of null, etc.)

4. x=number of characters advanced. Ratios are 4.2, 3.35, 3.03, 2.87, 2.77, 2.7, 2.65, 2.61 for x=0,1,2,...

As expected, the Commodorian guy writing that "score" report appends some acrobatic wording trick to make the C64 appear "faster". The "inverted dextral" is also possible on the Z80A: on the Zx Spectrum basic ROM you actually find it used for both keyword list and error text messages.

I guess that the Spectrum string routines timed by the Commodorian guy did not want *to use the "CPIR" Z80 instruction*

(compare, increment and repeat), analogue to the "LDIR" (load, increment and repeat) above, and surely did not want to use also the second set of Z80 registers. Above we saw a "ratio" of 1.6 when using "LDIR" instead of a wordy lenghty redundant routine (note: the latter also kicked in the ass the Commodore 64), so we can expect a large time improvement also in the strings routines figures of the Z80 by only using "CPI", "CPIR" and the extra registers.

5. Three separate Spectrum line routines were offered. All comparisons are of equivalent routines/ algorithms. The three routines are:

- slope<1, looped

- slope<1, unrolled across x-pixels, counting by columns
(more optimized version of the above)

- slope>1, unrolled across x-pixels
Spectrum times are Alvin's strange "average" cycles.

We wonder what are those "Alvin's strange average cycles". Why? What does it mean?

Drawing a graphics line on the screen using a Z80 does not require memory variables: that is, all data - deltas, pixel address, and so on - is contained in the Z80 registers (and almost one set remains free for other usage). No stack usage, no memory accesses except writing pixels. In contrast, 6510 has to use lots of memory variables. Did "Alvin" at least imitate the "DRAW" routine contained in the Zx Spectrum Basic ROM?

Another question: the Spectrum display map is not linear because of some optimization to draw text characters (remember, Zx Spectrum does not have the ugly "text without graphics" mode of the C64). I wonder how many times a screen-address recalc has been performed by someone unfamiliar with the Zx Spectrum display mapping. I guess that someone (especially Commodorians) may fall in that pitfall...

6. Using Ian Collier's revised algorithm.

7. "Average" cycle times consider one step in the x-direction followed by one step in x and y.

8. Looped version, slope>1 (no Spectrum version offered).

9. Spectrum times are modified by +7 if a-b<0, +3 if a+b>255, and +27 if placed in a subroutine (17 for CALL, 10 for RET). Cycle ratios assume inlined routine.

10. For multiplication of

```
constant*vector (i.e. matrix
multiply, or projection)
C64 version is 43 cycles for
first multiply and 25 cycles
for successive
multiplies. Thus, ratios are
around 3.1 and 3.4 for two
and three
successive multiplies.

11. Spectrum will choke,
badly.
```

The considerations in the note 9 and 10 seem to come from some bad programmer who did not know Z80 flags, mixing signed/unsigned data, index register usage, 16-bit pointer arithmetic, etc. Again, without access to the original code, we cannot make assumptions, except that the Spectrum routines could be a real mess because of the funny comments of the section above).

The consideration in the note 11 shows the Commodorians' Fundamental Approach to Evidence Reporting.

Again (again and again) we see that Commodorian fans:

- make their pro-C64 sermons using fake assumptions, unproven assertions, stupid slogans;
- make biased comparisons... sometimes involuntarily, because they try to apply to the Z80 their rough 6510 programming style;
- make always useless talk about "sophisticated chips", especially when that chips do not contribute for anything.

We also demonstrated what Commodorians do not want even want to hear:

- computationally, the Zx Spectrum is *always faster* than the Commodore 64;
- the Z80A clocked at 3.5 MHz outperforms the 6510 at 0.982 MHz even in those "major features" of the latter ("fast" memory access, "branching and jumping", and all that Commodorian fan sermons);
- the Z80A on the Zx Spectrum, with its large register set, large instruction set and high clock speed, definitively kicks in the ass the 6510 of the Slowcoach C64;

Alfonso Martone

Interface 1bis (Teil 1)



Nach längerer Wartezeit - bedingt durch den Deutschen Zoll - ist wieder mal ein feines Stück Hardware bei mir eingetroffen: das **Interface 1bis**. Die kleine Platine wurde in Südafrika entwickelt und gebaut. Hier eine kurze Beschreibung um welche Möglichkeiten das Interface 1bis den Spectrum erweitert:

•Joystick-Anschluss (Kempston kompatibel)

•Mouse-Anschluss (Kempston-

Maus kompatibel)
- micro SD-Karten Interface
- USB-Anschluss (zur Verbindung mit einem Windows-Rechner der als Server verwendet werden kann)
- der Befehlssatz ist kompatibel zum Interface 1

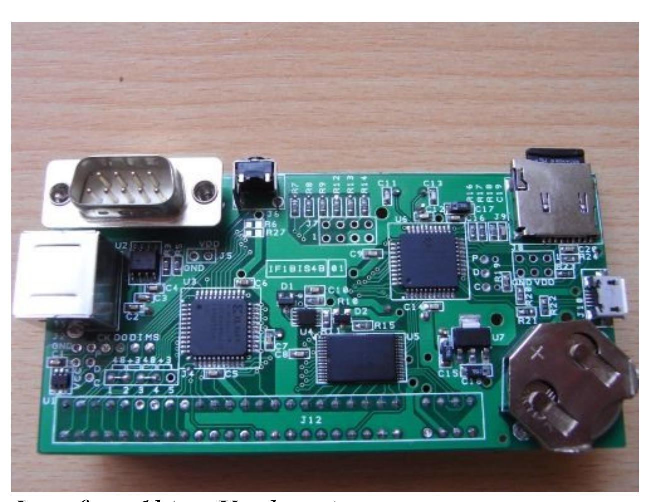
Das Interface kostet 46 € (inkl. Porto) - der Deutsche Zoll verlangt nochmal 19 % MwSt. Mit dabei ist eine Batterie und eine 4 GB SDHC Karte.
Das Interface ist hier erhältlich: 1IF1bis - nach Deutschland ist mit einer Lieferzeit von bis zu 4 Wochen (nach Versand) zu rechnen. Die Lieferung muss am zuständigen Zollamt abgeholt und die Rechnung vorgelegt werden (8,74 € Steuer sind noch zu bezahlen).
Hier noch die Übersetzung der englischen Seite:
Das Interface 1bis (Version4b) ist die einzige ZX Spectrum Erweiterungs-Karte welche die Kombination folgender Funktionalitäten bietet:
- lokale Speichermöglichkeit bis zu 128 GB, unterteilt in logische Laufwerk von 32 MB, auf einer micro SD Karte
- Remote-Speicher, von unbegrenzter Größe, auf einem Server über eine Full-Speed-USB-Port angeschlossen
- Kempston Joystick Anschluss
- PS/2 Maus Port welcher einen Kempston Maus oder Kempston Joystick-Ausgang bietet
- bereit für Ethernet - mittels eines optionalen Add-On Moduls von einem evt. Dritthersteller
- voll Software kompatibel zum original ZX Interface 1 von Sinclair Research Ltd. in Bezug auf die Microdrive Speicheroperationen
- erweitertes 'Extended BASIC' mit umfassenden Random-Access-Datei-Handling-Funktionen und voller Emulator-Dateien (.TAP und .Z80) Untersützung.

Hardware

Der Schlüssel zu seiner Vielseitigkeit ist die markante Architektur des IF1bis, die alle Peripherie-I/O-Operationen auf einen Mikrocontroller übergibt, der direkt an den Daten-Bus des Systems angeschlossen ist. Dies vereinfacht den Aufbau zu einem großen Teil auch in Verbindung mit einer rationalisierten Speicherkonfiguration - benötigt dafür nur einen 32KB nichtflüchtigen SRAM Chip - und fügt erhebliche Vorteile hinzu wie:
- kleines PCB (unter 0,5dm²)
- Null Schreibtisch Stellfläche durch vertikale Montage
- Einfaches, Zwei-Schicht-, PCB-Layout mit sehr kurzen Strecken, für reduzierte Geräuschentwicklung (Anm. der Red.: damit ist Rauschen in der Leitung gemeint).
- Geringe Leistungsaufnahme: bei unter 75mA, die niedrigste zwischen kompatiblen Geräten
- Geringe Buslast: unerreichte 0.7µA Datenbus Leckage und nur 0.5µA plus 1 (eins) Standard-CMOS-Last auf den anderen Linien
- Verbesserte Peripheriegerät Handhabung, da der Mikrocontroller mit seinen verschiedenen Hardware-Timern und komplexer Interrupt-Logik spezialisiert und besser für diese Aufgabe als die ZX Spectrum CPU ausgestattet ist
- Schnellere Datenübertragung

zwischen lokalen und Remote-Speicher, da es direkt vom Mikrocontroller über seine USB- und SPI-Schnittstellen unter Umgehung des ZX Spectrum durchgeführt wird

•Vereinfachtes Betriebssystem, das nur einen Gerätetreiber benötigt: für den "Slave Parallel Port“ des Mikrocontrollers, der wiederum alle niedrigen peripheren Steuerungsoperationen handhabt



Interface 1bis - Vorderseite

Das IF1bis weist einige Besonderheiten auf, die nicht auf vergleichbaren Geräten angetroffen werden. Bei der Optimierung der Kosten wurden keine Kompromisse hinsichtlich der Robustheit gemacht:

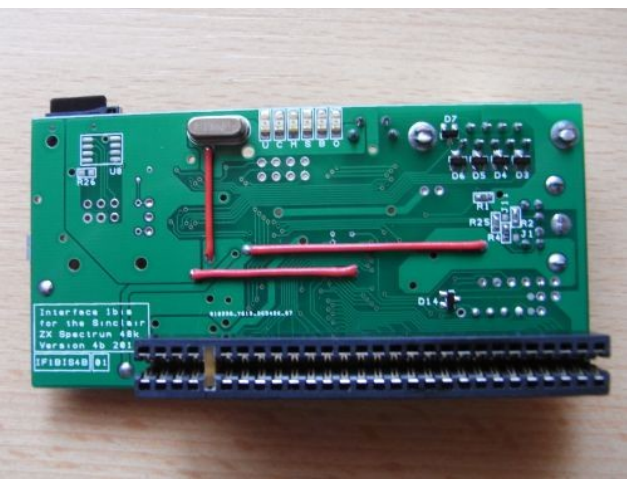
•Der nichtflüchtige SRAM wird durch einen einstellbaren, sehr präzisen, Spannungsüberwacher IC geschützt

•Der Steuerungs-Druckknopf ist entprellt und multiplexed wobei ein dedizierter Microcontroller benutzt wird, der saubere, digital gesteuerte RESET und NMI Signale erzeugt

•Die /ROMCS Linie wird nicht über Transistoren getrieben, sondern schnelle 3-State-Pufferum ausfallsicheres Speicher Paging sicherzustellen

•Der Joystick-Port ist vollkommen starr auf Metall Pfosten montiert

Und nicht zuletzt: Das IF1bis führt kein Hardware Trapping durch und funktioniert deshalb auch dann wenn der ZX Spectrum eine falsche/defekte M1 Line hat.



Interface 1bis - Rückseite

Betriebssystem

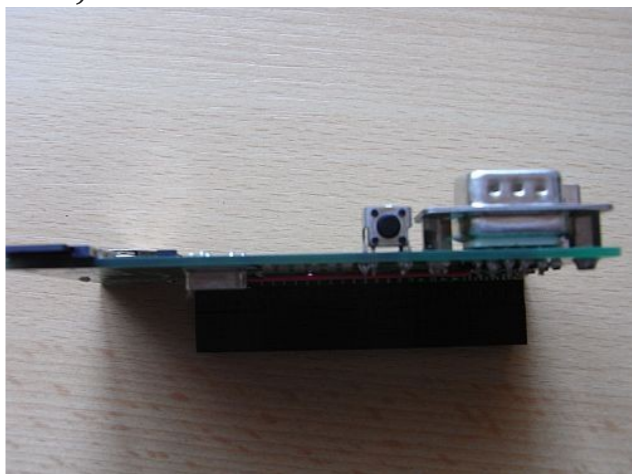
Das Interface 1bis ist, wie der Name vermuten lässt, zu 100% Software (abwärts) kompatibel zum Interface 1 von Sinclair Research in Bezug auf die Microdrive Speicheroperationen, sowohl was die Basic Befehle als auch den 'hook code' Level betrifft.

Die ZX Net und RS232-Schnittstellen Kommunikations-Funktionen des Originals wurden nicht als solche umgesetzt, da sie angesichts ihrer begrenzten Funktionalität und Peer-to-Peer-Natur als unpraktisch gelten. An ihrer Stelle wurde ein Remote-Datei-System für ein Client-Server-Netzwerk über einen USB-Anschluss entwickelt.

Alle Datei-Handling Operationen können auch auf beiden implementierten Dateisystemen durchgeführt werden:

- •lokal, für die SD-Karte (Gerät "M") und
- •remote, für den Datei Server (Gerät "V")

mit der Ausnahme die Server-Festplatte formatieren zu können, was aus offensichtlichen Gründen unmöglich ist :-)



Interface 1bis - Button und Kartenslot

Die wichtigsten Verbesserungen am erweiterten BASIC des ursprünglichen ZX Interface 1 sind:

- •die Dateigrößen Begrenzung ist 16MB
- •sequentieller Zugriff ist nicht auf 'PRINT' Dateien begrenzt
- •sequentieller Zugriff auf "M-Kanäle" kann selektiv für Input und Output geöffnet werden
- •Daten im Speicher können gelesen, geschrieben oder verifiziert werden, durch direkten zufälligen Zugriff, zu (oder von) einer angegebenen Position in einer Datei, eines beliebigen Typs, ohne die Notwendigkeit, ihn zu öffnen
- •beide sequentiellen Zugriffs-Zeiger von "M-Kanälen" und zufällige Zugriffs-Zeiger von "file handles" können an jede Position verschoben werden
- •volle Untersützung für die folgenden Emulator-Dateien: .TAP Tape Images, .Z80 Snapshots und .SCR Screen Dumps
- •abgekürzte Syntax, welche den Wegfall von 'device type' und 'device number' in einem Statement erlaubt, wenn sie die gleichen sind wie im vorigen Statement

Das Betriebssystem des Interface 1bis hat eine sehr lange Entwicklungsgeschichte, geschrieben vor 18 Jahren und seitdem ständig verbessert und an neue Hardware-Konfigurationen angepasst.

Es ist ein ausgereiftes, stabiles Produkt, zugeschnitten auf die praktischen Bedürfnisse und bietet wohl eine einzigartige Funktionalität, Geschwindigkeit und Benutzerfreundlichkeit.



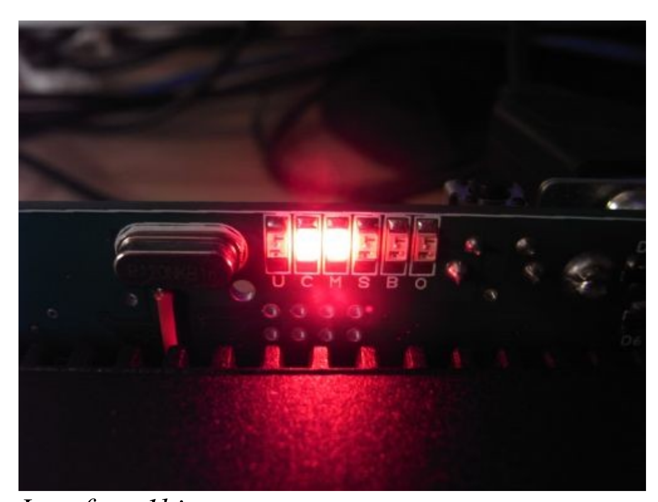
Interface 1bis - PS2-Anschluss

Anschluss und weitere Hinweise (Bedeutung der LEDs)

Der Anschluss ist wie bei allen Erweiterungen: Interface hinten am

Spectrum anschließen - natürlich nur im ausgeschalteten Zustand.
Das Interface 1bis hat einen kleinen Taster der für die Funktion sehr wichtig ist. Dieser Taster hat mehrere Funktionen:

- •ON/OFF Wenn er länger als 1,2 Sekunden gedrückt wird, wird der Zustand des Interfaces von 'OFF' in 'ON inaktiv geändert.
- •RESET Wenn er zwischen 0,5 und 1,2 Sekunden gedrückt wird, wird ein 5ms-wide, active-low Puls auf der Reset Line des ZX Sepctrums generiert.
- •NMI Wenn er kürzer als 0,5 Sekunden gedrückt wird, wird ein 5ms-wide, active-low Puls auf der NMI Linie generiert.

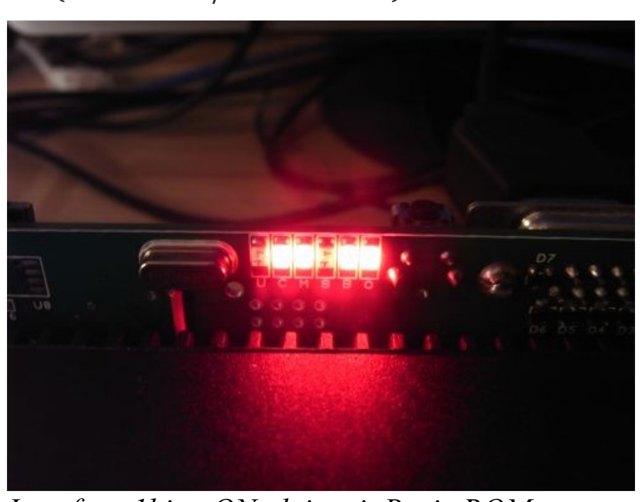


Interface 1bis

Was mir am Anfang des öfteren passiert ist: Ich habe einen Befehl eingegeben und es sind Fragezeichen erschienen. Dann ist mir aufgefallen das zwar einige LEDs leuchten aber nicht die wichtigen beiden: O und B. Daher den Taster für etwas mehr als 1,2 Sekunden drücken und gleich nochmal kurz für das BASIC - schon klappen die Befehle.
Hier nun deshalb nochmal die Bedeutungen der LEDs:

- •LED 'O':der 'ON/OFF' Status des Interfaces
- •LED 'B': das 'BASIC ROM' des Interface ist paged-in
- •LED 'S': das 'Shadow ROM' des Interface ist paged in
- •LED 'M': wenn es leuchtet: die Maus ist als 'Kempston' Maus aktiv -wenn es blinkt: die Maus ist als 'Kempston' Joystick aktiv
- •LED 'C': die SD-Karte wurde identifiziert
- •LED 'U': die Verbindung zum Server wurde hergestellt

Nach dem Einschalten ist das Interface 1bis wie oben beschrieben im OFF Status (O LED ist aus). Wenn das O-LED leuchtet reagiert das Interface auf I/O Anfragen aber es läuft das BASIC des 48K Spectrum. Der Joystick und der Maus-Anschluss sind aber funktionsfähig. Die 128er Spectrum arbeiten nur im 48K Modus mit dem Interface (+2 und 128K Toastrack getestet).
Wenn das O-LED leuchtet ist das Interface voll funktionsfähig und der Spectrum arbeitet mit dem Extended Basic des Interface. Das B LED und das S LED signalisieren welches ROM aktiv ist (B=BASIC/S=Shadow).


Interface 1bis - ON aktiv mit Basic ROM

Benutzung der SD-Karte

Die SD-Karte sollte im ausgeschalteten Zustand eingelegt werden. Wenn die Karte erkannt wurde, leuchtet nach dem Einschalten das C______LED. Im Gegensatz zu anderen Lösung bei den Speccys, kann diese SD-Karte nicht einfach am PC befüllt werden!

Befehle
FORMAT

```
FORMAT „m“;0
```

Damit wird nicht wirklich etwas formatiert oder gelöscht ;-) Die SD-Karte muss auch nicht partitioniert werden, da sie automatisch in bis zu 255 logische Laufwerke unterteilt wird, die alle 32MB groß sind. Diese Laufwerke müssen vor der Benutzung formatiert werden:

```
FORMAT „m“; d
```

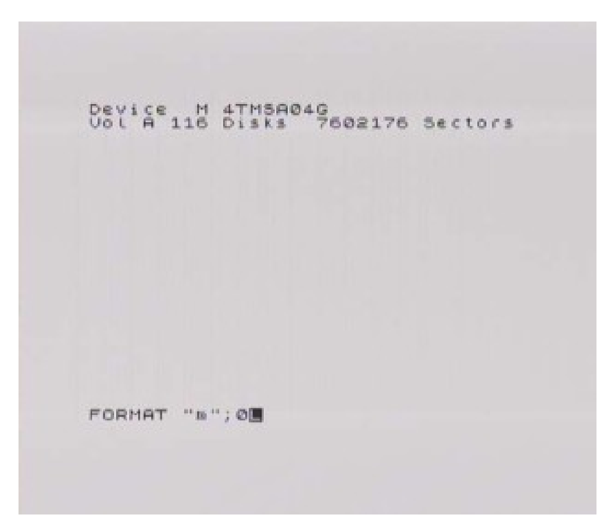
d ist die Nummer des Laufwerks! Bei einer 1GB Karte sind es ungefähr 58 Laufwerke - ich denke man kann sich hier ein kurzes BASIC-Programm schreiben, welches das Formatieren übernehmen kann ;-)
Wenn die Karte größer als 8GB ist, wird es etwas komplizierter - Dann gibt es mehrere Volumes auf einer SD-Karte (diese werden mit A bis O bezeichnet). Der Name des Volumes folgt nach dem Gerätenamen:

```
FORMAT „me“;22
```

Hier würde auf Volume "E" das Laufwerk 22 formatiert..... Außerdem können noch die Zuordnungseinheiten eingestellt werden:

```
FORMAT #c;“m“;d
```

„c“ ist die Zuordnungseinheit – 2, 4, 8, 16
Zum Glück hat der Entwickler schon zwei Programme geschrieben um das Leben mit der SD-Karte etwas komfortabler zu machen. Zum einen gibt es das Programm "COPYTREE" mit dem man schnell Daten/Verzeichnisse kopieren kann. Noch wichtiger ist der "Game Launcher", den es auf der Homepage zum Download gibt. Für diesen ist allerdings ein Update der Firmware nötig ;-) Wenn man die Datei heruntergeladen hat, muss sie am besten auf Laufwerk 1 der SD Karte gespeichert und in RUN umbenannt werden (Kopie vom Server-Laufwerk - siehe Teil 3). Damit kann das Programm dann einfach mit "RUN" geladen und gestartet werden. Leider ist die Länge der Dateinamen auf 10 beschränkt - sollten sie länger sein, werden die Namen gekürzt. Damit kann man dann einfach zwischen den Geräten, Laufwerken und Verzeichnissen wechseln und natürlich Programme starten!

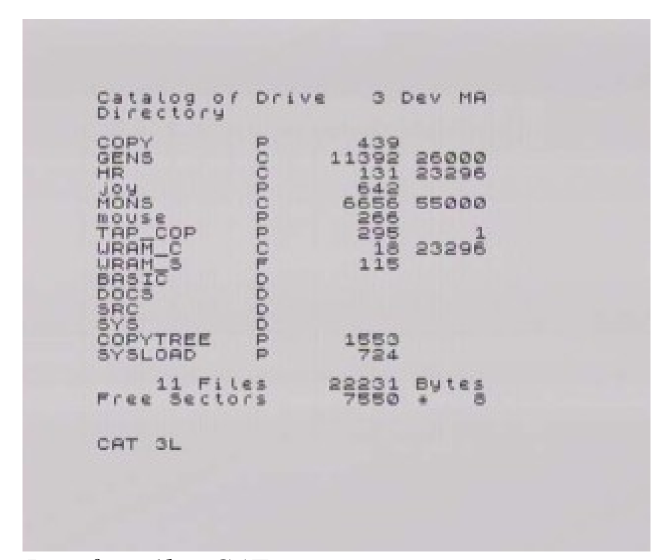


Interface 1bis FORMAT

CAT

```
CAT „m“;d
CAT d
CAT
```

Damit wird das Inhaltsverzeichnis der SD-Karte angezeigt, wobei "d" wieder für das Laufwerk steht.
Die Dateinamen erscheinen in der ersten Spalte der Ausgabe durch den CAT Befehl, während die zweite Spalte die Dateiendungen anzeigt:



Interface 1bis CAT

- P - BASIC Programm (.ZZP)
- N - Nummern Array (.ZZN)
- A - String Array (.ZZA)
- C - Code Block (.ZZC)
- F - PRINT File (.ZZF)
- X - Text File (.TXT)
- B - Binäre Datei (.ZZB)
- S - Screen Dump (.SCR)
- T - Tape Datei (.TAP)
- Z - Snapshot (.Z80)
- D - Directory

Die dritte Spalte zeigt die Dateigröße an und die vierte:

- die Startzeilennummer von BASIC Programmen
- den Array Namen oder
- die Lade Adresse von „CODE“ Blöcken

Der freie Speicher auf dem logischen Laufwerk wird als Anzahl von Zuordnungseinheiten (Clusters) angezeigt.

Verbindung mit einem Server

Eines der Highlights des Interface 1bis ist die Verbindung per USB-Kabel mit einem Windows-Rechner. Dafür werden 3 Sachen benötigt:

- ein passendes USB-Kabel (auf der Interface-Seite muss es ein Mini-USB-Stecker sein)
- ein Treiber: IF1bis.inf (auf der Homepage des Projekts zu finden)
- das "Server"-Programm: ZXS32.exe (ebenfalls auf der Homepage zu finden)



Interface 1bis - Gerätemanger

Als erstes einfach mal den Spectrum mit dem Interface 1bis einschalten und das USB-Kabel mit dem Windows-Rechner verbinden. Es sollte nun die Hardware-Erkennung starten. Danach landet das Interface erst Mal unter "andere Geräte". Nun muss einfach der Treiber aktualisiert und dazu der vorher heruntergeladene Treiber F1bis.inf verwendet werden. Danach war es bei mir immer so, dass das Gerät dann als nicht funktionsfähig angezeigt wurde. Das Problem kann einfach beseitigt werden, indem es im Gerätemanager von Windows einmal deaktiviert und wieder aktiviert muss - danach funktioniert es.

WEISST DU MIT WEM SICH DEIN SPECTRUM HEUTE NACHT TRIFFT?

???



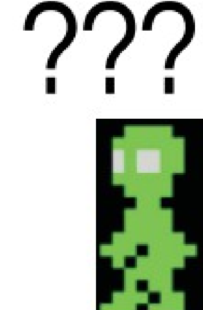
???

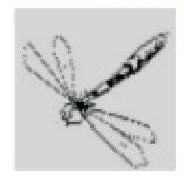


???



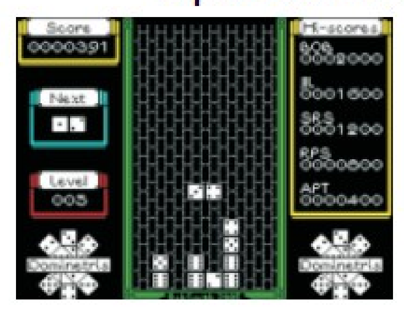


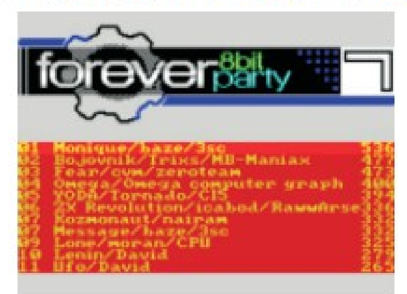




SCENE +

Spectrum-Entertainment starts here...







Einziges Spectrum Tape/Disk/Tap-Magazin
+D/Disciple, Opus, MB02, Tape oder .TAP Download
in Deutsch und Englisch

Erhältlich vom:

Scene+ Redakton, Mirko Seidel Neenstetter Str. 20, 89183 Breitingen
http://www.speccy-scene.de/ e-mail: ms-256-email@gmx.de
EIN MAGAZIN DES SPECTRUM-USER-CLUB, GERMANY